

REMARKS

Claims 1 and 49 have been rewritten as claims 56, and 57.

In one embodiment of the present invention, as set forth in claim 56, a method for generating test scripts provides at least one particular test case in a data store in an abstract representation form. Abstract representations are used that have at least three separate components including at least one application state, at least one external interaction sequence, and at least one input data. The application state has at least one of: (a) a set of application objects associated with a set of attributes and their values, or (b) a runtime snapshot of an application under test which defines a context of external interaction. At least one test case is selected in its abstract representation using rules for the selection of the application state, the external interaction sequence and the input data. The rules are used to validate the rule-based test case against an application object model. The application object model is a metadata representation for modeling application under test and includes components selected from application object type definitions for application objects, attribute definitions for each application object type, definitions of methods and events that are supported by each application object type and definitions of effects of events on an application state. At least one test script is generated based on the rule-based selected. The test script can be run in a plurality of target test execution environments.

Support for the "separate components" is found in Figure 3

Support for "rules are used for the selection of the at least one application state, the at least one external interaction sequence and the input data" is found in paragraph 0049 of the specification.

Support for "application object model and its components is found in paragraph 0035 of the specification and in Figure 6.

With the '564 Patent, the steps and procedures encoded into the test cases utilize terminology, test strings and/or nomenclature native to an automation tool so that the steps and procedures can be converted into an automation tool recognized test scripts. With the '564 Patent, there is no abstract representation of test cases or a conversion of original test cases/scripts into an abstract representation.

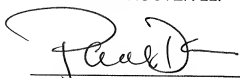
With the present invention, a test case is selected in its abstract representation using rules for the selection of the application state, the external interaction sequence and the input data. The rules are used to validate the rule-based test case against an application object model. Again, the '564 does not use abstraction representations and does not use rules as with the present invention.

With WinRunner, the GUI map is not an abstract representation. WinRunner is specific to a UI. The WinRunner GUI map represents GUI objects. The abstract representation of the present invention is a representation of the overall test case, including test steps, state, which include objects, and test data. Rules are used for the selection of the application state, the external interaction sequence and the input data. The rules are used to validate the rule-based test case against an application object model. WinRunner does not use abstraction representations and does not use rules as with the present invention.

CONCLUSION

It is submitted that the pending claims of the present application as amended and presented herein are in form for allowance, and such action is respectfully requested. The Commissioner is authorized to charge any additional fees which may be required, including petition fees and extension of time fees, to Deposit Account No. 07-1700 (Docket No. SYM-0004).

Respectfully submitted,
GOODWIN PROCTER LLP



Paul Davis, Reg. No. 29,294

Date:

8/15/08

135 Commonwealth Drive
Menlo Park, CA 94025
Tel: 650/752-3106
Customer No. 77845